



Aras Innovator 31

Life Cycles

Document #: D-008969

Last Modified: 1/23/2024

Copyright Information

Copyright © 2024 Aras Corporation. All Rights Reserved.

Aras Corporation
100 Brickstone Square
Suite 100
Andover, MA 01810
Phone: 978-691-8900

E-mail: support@aras.com

Website: <https://www.aras.com/>

Notice of Rights

Copyright © 2024 by Aras Corporation and/or its affiliates. All rights reserved.

This document is protected by U.S. and international copyright laws and conventions. No copyright may be obscured or removed from this document. This document may not be modified or altered, or reproduced or transmitted in any form, without the explicit permission of the copyright holder.

Aras Innovator, Aras, and the Aras Corp "A" logo are registered trademarks of Aras Corporation in the United States and other countries.

All other trademarks referenced herein are the property of their respective owners.

Notice of Liability

THIS DOCUMENT IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY, AND THE CONTENTS HEREOF ARE SUBJECT TO CHANGE WITHOUT NOTICE. THE INFORMATION CONTAINED IN THIS DOCUMENT IS DISTRIBUTED ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR A WARRANTY OF NON-INFRINGEMENT. ARAS SHALL HAVE NO LIABILITY TO ANY PERSON OR ENTITY WITH RESPECT TO ANY LOSS OR DAMAGE CAUSED OR ALLEGED TO BE CAUSED DIRECTLY OR INDIRECTLY BY THE INFORMATION CONTAINED IN THIS DOCUMENT OR BY THE SOFTWARE OR HARDWARE PRODUCTS DESCRIBED HEREIN.

Table of Contents

Send Us Your Comments	4
1 About Life Cycles.....	5
2 Creating a Life Cycle Map	6
2.1 To create a Life Cycle Map	6
2.2 State Property	7
2.3 Creating A Transition	8
2.3.1 Life Cycle Transition Methods.....	8
3 Configuring Email	9
3.1 Creating a Configured email	10
3.2 Using Variables in Email Text	11
4 Item Behavior	13
4.1 Behavior When Parent (the Source Item) Is Versioned First.....	13
4.2 Behavior When Child (the Related Item) Is Versioned First	15

Send Us Your Comments

Aras Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for future revisions.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where and what level of detail?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, indicate the document title, and the chapter, section, and page number (if available).

You can send comments to us in the following ways:

Email:

TechDocs@aras.com

Subject: Aras Product Documentation

Or,

Postal service:

Aras Corporation

100 Brickstone Square

Suite 100

Andover, MA 01810

Attention: Aras Technical Documentation

If you would like a reply, provide your name, email address, address, and telephone number.

If you have usage issues with the software, visit <https://www.aras.com/support/>

1 About Life Cycles

A Life cycle is a series of states (i.e., stages or gates) that an Item instance traverses during its existence. Most business processes define high-level stages to track the progress of an object in a Life cycle. The following is an example of a life cycle:

The screenshot displays the 'Life Cycle Map 1' configuration window. It features a menu bar (File, Edit, Views, Actions, Reports, Tools, Help) and a toolbar with various icons. The main area is divided into several sections: a left sidebar for 'Name' and 'Description'; a central 'State' tab with 'General' settings (Name, Start, Label, Image) and 'Item Settings' (Released, Not Lockable, Item Behavior, State Permissions, Workflow, History Template); and a bottom section with a 'start' button and a 'Start' label.

A Life Cycle consists of States, which are basically a series of actions and steps, and Transitions, which are basically paths between the different states. Let's go through this cycle together to gain some basic understanding.

When an instance of the ItemType Vendor is created, named PartsVendor1, it is immediately placed into the Preliminary state. Usually, to qualify as an Approved Vendor, the vendor must meet some company-specific criteria, or go through a specific process. When this criterion is met, a member of the Component Engineering Identity can then promote PartsVendor1 to the Approved state. If PartsVendor1 satisfies even stricter criteria (usually tighter delivery dates, or volume discounts), then a member of the Component Engineering Identity can promote this vendor to the Preferred state. Of course, if the vendor fails in some way, again according to the company's set criteria, then the vendor can be transitioned to the Inactive state. Typically, within each state, there are a series of steps or actions that need to be taken by prescribed Identities. These are usually described and assigned to a Workflow map, which can be automatically activated when an item enters a specific life cycle state.

As a very powerful tool, there can be all sorts of settings activated and deactivated triggered on the item's state change. For example, one setting could be that when someone from the Procurement Identity selects a vendor for a specific part, only the Vendors in the Preferred state is used to populate the list of available vendors. Another feature is that when an item enters any Life Cycle state, an e-mail notification can be set to automatically inform the selected Identities of any action items that must be taken because of this state change.

Different Life Cycle Maps can be assigned to different Classes. See Class Structure

2 Creating a Life Cycle Map

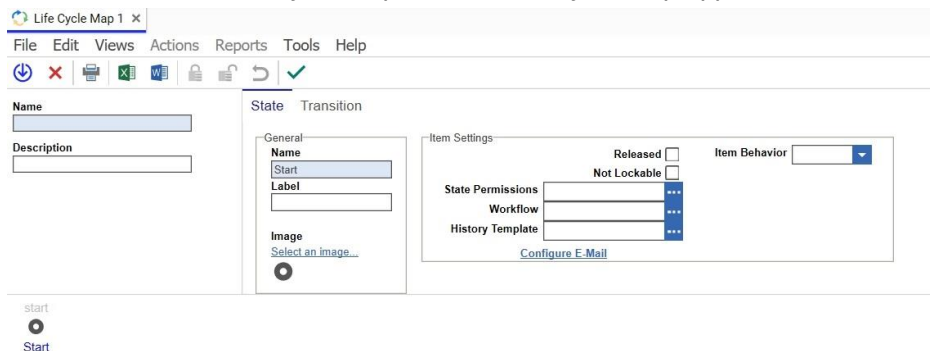
A Life Cycle Map can be created independently of any ItemType to which it may be attached later. As a matter of fact, many ItemTypes can use the same Life Cycle map. Additionally, an ItemType may contain multiple Life Cycle Maps which are tied to the ItemType's Classification Hierarchy. When an item instance is created for an ItemType, Aras Innovator chooses the appropriate LifeCycle Map based on the initial classification of the Item. When an item instance is updated, Aras Innovator checks for any changes in the classification property. If the item's class has changed to an allowed class that has a different Life Cycle Map and the item's current Life Cycle state is still in the 'Start State', then Aras Innovator will set the Item's Life Cycle to the one specified for the new Classification. If the Classification is changed for an existing Item Instance which has a Life Cycle that is not in the 'Start State', then Aras Innovator will reject the Classification change to avoid interfering with the normal progression of the Life Cycle. If required, the Life Cycle Map can be modeled to allow for promotions back to the Start State so that normal users can perform a class change.

2.1 To create a Life Cycle Map

1. In the Navigation pane, select Administration>LifeCycle Maps. The following menu appears:



2. Click Create New Life Cycle Map. A blank LifeCycle Map appears:



3. Fill in the Life Cycle Map properties:
 - Name - the name of the map
 - Description- the description of the map
4. Create a new State by right-clicking anywhere on the map canvas and selecting Add State from the popup menu. A new State is created and remains selected by default.

2.2 State Property

State Property	Definition
Name	The name of the state
Image	Click the Select an image link to access the Image Browser dialog. Select an icon for the state, then click the Return Selected button  to complete the selection.
Released	This property applies only if this Life Cycle map is associated with an ItemType that is versionable. A Boolean to indicate that the status of this state is Released. Items in a state with a Released status are moved back to the Start state upon the next "Lock/Unlock/Edit" sequence, and the Major Rev property is incremented. Only one state per Life Cycle can be marked as Released.
Not Lockable	A Boolean: when set to true, the item in this state cannot be edited. Typically, this property is set to true when the status of the state is Released. To edit the item, it has to be promoted to a new state, which would typically (depending on the life cycle) create a new version.
Item Behavior	Applicable only if the item going through this life cycle state contains one or more relationships with versionable items. There are four types of Item Behavior, and they are described below. Be aware that the Behavior set by the RelationshipType influences the Behavior set by the state as well. To see a full explanation of how these behaviors interact together, see Item Behavior. Fixed - The source item (in the current state) points to a specified generation of the related item. Float - The source item (in the current state) points to the latest generation of the related item. Hard Fixed - The behavior is Fixed, as described above, and cannot be changed by any subsequent states in this life cycle. Hard Float - The behavior is Float, as described above, and cannot be changed by any subsequent states in this life cycle.
State Permissions	A set of permissions may be defined for each state, which will govern the access to the item while it is in this state. The state permissions override any other permissions set on the Item. Promoting to a state changes the Item permissions to match the state permissions. If no state permissions are set, then the Item permissions are not changed. For an example of state permissions, look at the default Part Life Cycle map. While a Part is in the Preliminary state All Employees can get and update it. However, once the Part enters the Review state, only members of the Aras PLM and the Administrators identities can update it.
Configure E-Mail	Click on this link to configure e-mail, to notify specified identities that the item has entered this state.

Workflow	Click the  button to the right of the <i>Workflow</i> field. Select an <i>allowed</i> Workflow from the pop-up search dialog. The selected Workflow will be initiated when the Life Cycle State is activated. Note: An <i>allowed</i> Workflow is one that is related to the same ItemType to which the Life Cycle is also related.
Label	This field defines the label that is displayed within the system for this Life Cycle State. If Aras Innovator is configured for Multilingual support, then this field allows the label to be defined in multiple languages. To set a multilingual value for this field, click the  button to the right of the <i>Label</i> field. The system will pop-up a dialog for inputting a value in different languages.
History Template	Click the  button to the right of the <i>History Template</i> field. Select a History Template from the pop-up search dialog.

2.3 Creating A Transition

1. Right click on the state to create a transition.
2. Select Add Transition from the popup menu. A transition is created from the specified state and connected to the mouse. Drag the mouse to the state to connect and click to connect the transition.

2.3.1 Life Cycle Transition Methods

Life Cycle Transitions can be configured with **pre** and **post** methods.

The pre-methods are executed before the transition to the new state takes place. They are commonly used for validation purposes. If the pre-method fails or throws an error, the transition does not happen.

The post-methods are executed after the transition is completed, and they include two options that can be configured during Life Cycle setup: **in-scope** and **out-of-scope**.

By default, post methods are configured to be **out of-scope**. This means that if the method fails during execution, the transaction is not rolling back.

Configuring the post-method to be **in-scope** means that if the method fails, the transaction is going to roll back to the starting stage.

3 Configuring Email

Email is used to notify the selected identities that an item has either entered a specific life cycle state, or that it is in transition between states. So, for both the state and the transition, the configuration dialog is the same.

Configure E-Mail

E-Mail Message

Hidden

Name	From User [...]	Subject	Body Plain [...]	Body Html [...]	Query String [...]	Of Itemtype [...]
SSVC Notifica...	Innovator Ad...	Visual Collab...		<table border...		

< Prev Next > Page: 1 of 1 | 1 Results

The top part of the dialog allows for the message itself to be specified. Users can have more than one message specified, if users want a different message to be sent to different identities. In the bottom part of the dialog, specify which identities will receive the message currently selected in the top pane.

Selecting a configured e-mail:

1. Click the Configure E-Mail link in the Life Cycle map selected. The Configure E-Mail dialog box appears.

2. Click the Add E-Mail message button . The Email Message search dialog appears.

3. Click the Run Search button to see a list of previously configured emails.

Select Items

E-Mail Message

Simple

Name	From User [...]	Subject	Body Plain [...]	Body Html [...]	Query Strin...
SSVC Notific...	Innovator Ad...	Visual Collab...		<table border...	
Express EDR...	Innovator Ad...	\$(Item2)ite...		<BODY><font...	<Item type="...
CM Activity N...	Innovator Ad...	New Innovat...		<BODY><font...	<Item type="...

< Prev Next > Page: 1 of 1 | 3 Results

4. Select a message type and click . The pre-configured email is added to the list. A new line is created in the top pane. The following table describes the properties:

Property	Description
Name	The name of the e-mail message itself
From User	The identity from whom the message will be sent. If none is specified, the identity configuring the e-mail notification will be used. Hit F2 to bring up a search dialog with all available identities. The chosen identity must have a valid e-mail address.
Subject	The subject line of the e-mail.

Body Plain	Plain text version of the message to be sent (for recipients who do not accept HTML)
Body HTML	HTML version of the message to be sent
Query String	If a user wants to write complex notifications, that require data from various items in Innovator, a query can be written to request this data from the database. The retrieved data can then be referenced in the body of the e-mail message. For example, to show values of a related item in the e-mail message, a query will be writtento retrieve it. To learn how to write these queries, please refer to the Advanced Programming course, or contact an Aras Consultant.

- Click the Add Identities button in the Identities section of the dialog to specify existing identities that will receive the e-mail (make certain that those identities have a valid e-mail address). Click the New Identity to add an identity. The Identity search dialog appears.

Select Items

Identities

Simple

Name	Description	Is ...	Classification

< Prev Next > Page: 1 25

OK Cancel

- Select the identities to receive the email and click  to add them.

3.1 Creating a Configured email

Use the following procedure to create an email:

- Click on the Configure Email link in the LifeCycle Map that have been created/selected. The Configure E-Mail dialog appears.
- Click the New Email button  to create an email. A row is added to the dialog box.

Configure E-Mail

E-Mail Message

Hidden

Name	From User [...]	Subject	Body Plain [...]	Body Html [...]	Query String [...]	Of Itemtype [...]
SSVC Notifica...	Innovator Ad...	Visual Collab...	-table border...			

< Prev Next > Page: 1 of 1 4 Results

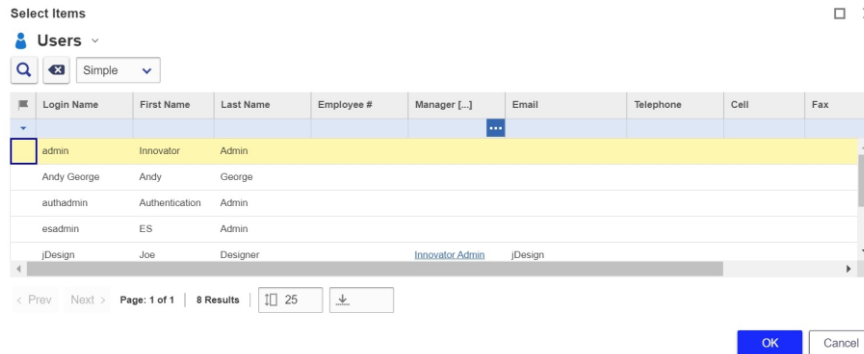
Identities

Hidden

Name

< Prev Next > Page: 1 of 1 0 Results

- Enter a name for the email in the Name field.
- Double-click the From User cell and click the ellipses button  to select the users to associate with the email. The User search dialog appears.



Select Items

Users

Simple

Login Name	First Name	Last Name	Employee #	Manager [...]	Email	Telephone	Cell	Fax
admin	Innovator	Admin						
Andy George	Andy	George						
authadmin	Authentication	Admin						
esadmin	ES	Admin						
jDesign	Joe	Designer		Innovator Admin	jDesign			

OK Cancel

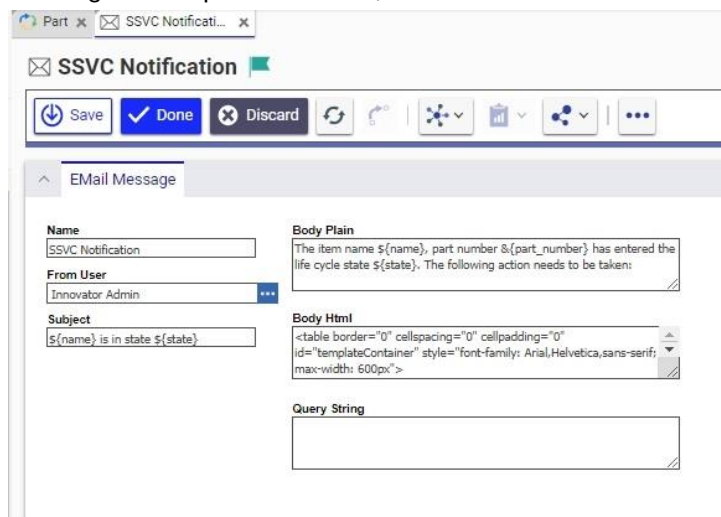
- Select usernames to associate with the email and click .

3.2 Using Variables in Email Text

Users can use variables in the e-mail text to identify the exact item and its property values, making the e-mail message more specific. Users can insert any item property using the syntax `#{property name}`, including system properties. There are also the following runtime variables supported in Innovator:

Variable	Definition
<code>#{USER}</code>	The current logon name
<code>#{ALIAS}</code>	The alias identity of the user
<code>#{DATE}</code>	String containing today's date
<code>#{TIME}</code>	String containing the current time

Here is an example. We have an item with the following properties: name, part_number, and color. It also has the usual system properties, and we will use the property state to indicate which life cycle state the item is entering. An e-mail message could be configured as shown below. If a user right clicks on the e-mail row and select View "Email Message" from the pop-up menu, an email message in a separate window, as shown here.



Part x SSVC Notificati...

SSVC Notification

Save Done Discard

Email Message

Name: SSVC Notification

From User: Innovator Admin

Subject: #{name} is in state #{state}

Body Plain: The item name #{name}, part number #{part_number} has entered the life cycle state #{state}. The following action needs to be taken:

Body Html: <table border="0" cellspacing="0" cellpadding="0"> id="templateContainer" style="font-family: Arial,Helvetica,sans-serif; max-width: 600px">

Query String:

Notice the use of two variables in the subject line - \${name} and \${state}. Please make sure to use the property names, not labels. The Body of the message is using the additional property \${part_number}.

4 Item Behavior

Item Behavior is a property of the life cycle state that sets the behavior of the connection between the source item, or the parent, and the related item, or the child. This property is only applicable if the source item and the related item are versionable. Also, keep in mind that there is also an Item Behavior property defined on the RelationshipType itself (see RelationshipType Item Behavior). These two properties - the Item Behavior of the RelationshipType and of the Item Behavior of the Life Cycle State - work together to control the configuration of the source and related items.

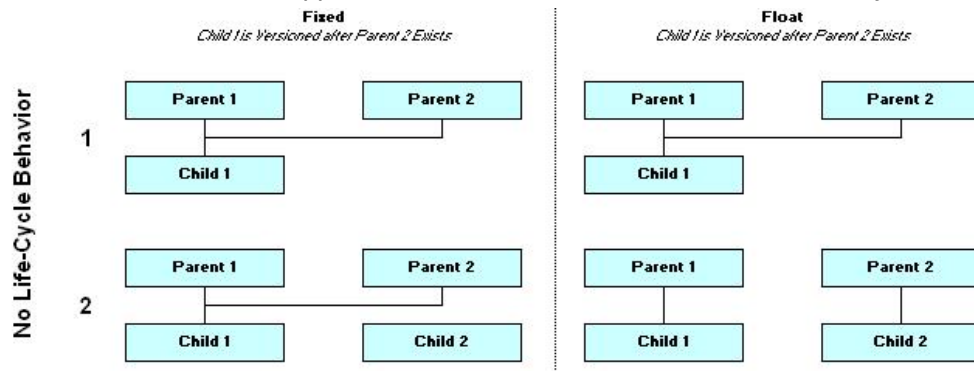
The settings Hard Fixed and Hard Float mean different things for the RelationshipType behavior and the Life Cycle State behavior. The table below explains these settings:

	Hard Float	Hard Fixed
RelationshipType	Float behavior that cannot be modified by the life cycle state settings	Fixed behavior that cannot be modified by the life cycle state settings
Life Cycle State	Float behavior that is in effect until the life cycle ends, and cannot be changed by subsequent life cycle states	Fixed behavior that is in effect until the life cycle ends, and cannot be changed by subsequent life cycle states

Furthermore, there are two categories to consider - what happens to the items if the child (or the related item) is versioned first, as opposed to the parent (or the source item) being versioned first. Let's look at the first case first.

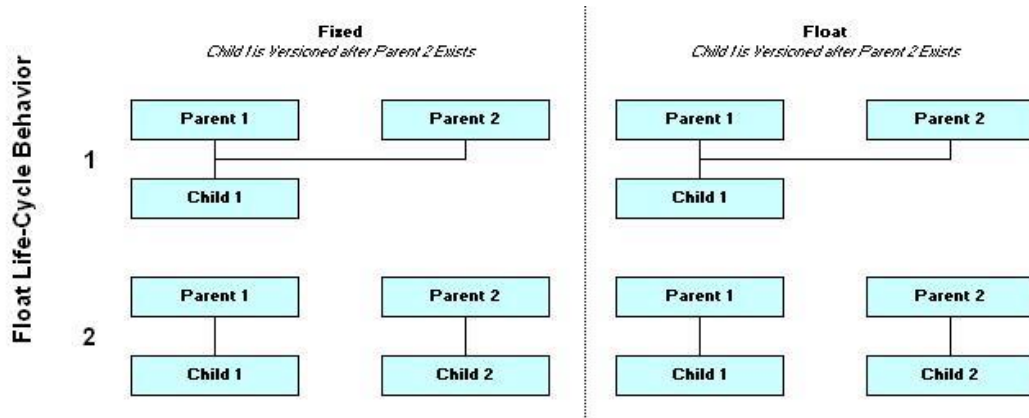
4.1 Behavior When Parent (the Source Item) Is Versioned First

Let's first consider what happens when there is no behavior set on the life cycle state:

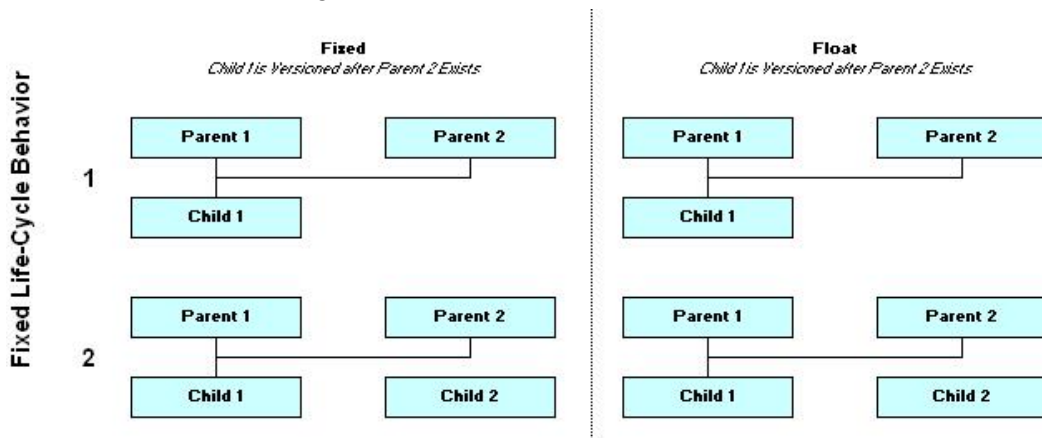


Notice that with the Float RelationshipType behavior, when Child2 is created, Parent1 still points to Child1. The reason for this is that when a new version of the Parent is created, the previous versions are then Hard Fixed in their current configuration. So, when Parent2 is created Parent1 is pointing to Child1. It remains in that state going forward, no matter how many times Child is versioned after that.

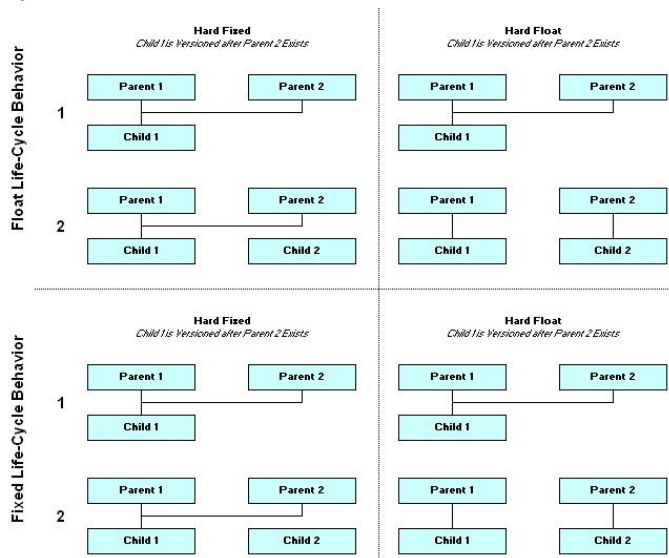
When the Life Cycle Behavior is set to Float, it forces even the RelationshipType defined as Fixed to act as Float. See the figure below.



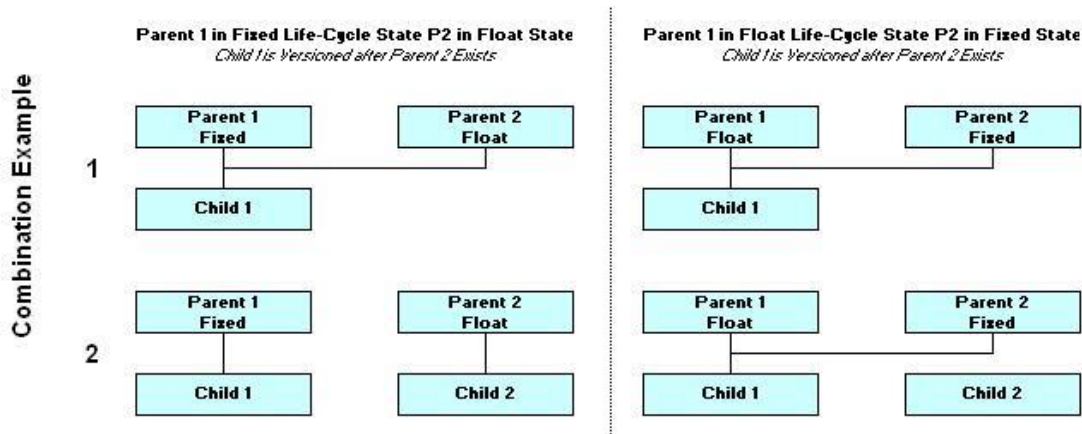
When the Life Cycle Behavior is set to Fixed, it forced even the RelationshipType defined as Float to act as Fixed. See the figure below:



When the RelationshipType is defined as Hard Fixed or Hard Float it overrides whatever the behavior is set by the Life Cycle state. Notice the two figures below showing that the behavior defined by the RelationshipType rules the behavior, regardless of the behavior defined by the Life Cycle state.



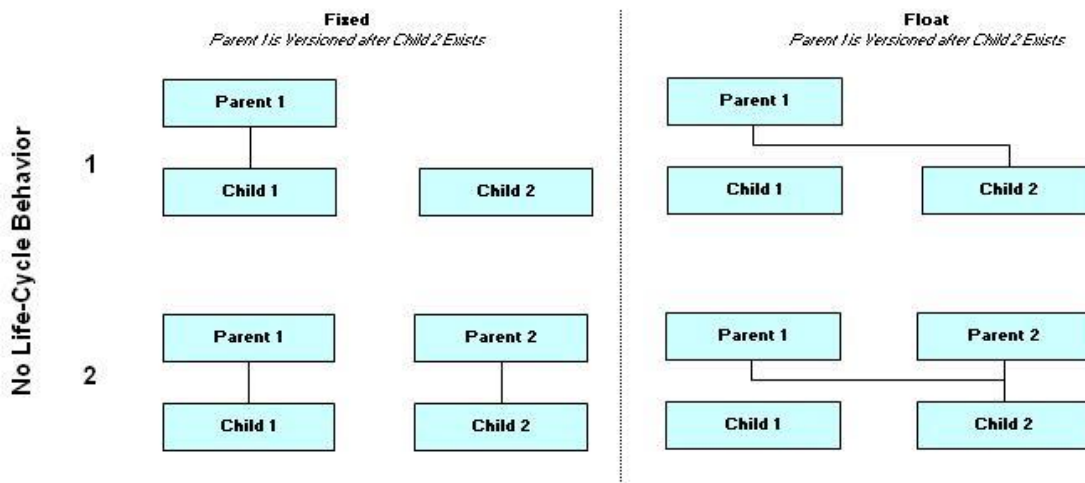
If the RelationshipType Behavior is not set to Hard, it can be overwritten by the Life Cycle State Behavior. The figure below shows what happens when one generation of the parent is in one state, with one type of behavior, and the next state is in another life cycle state, with a different behavior.



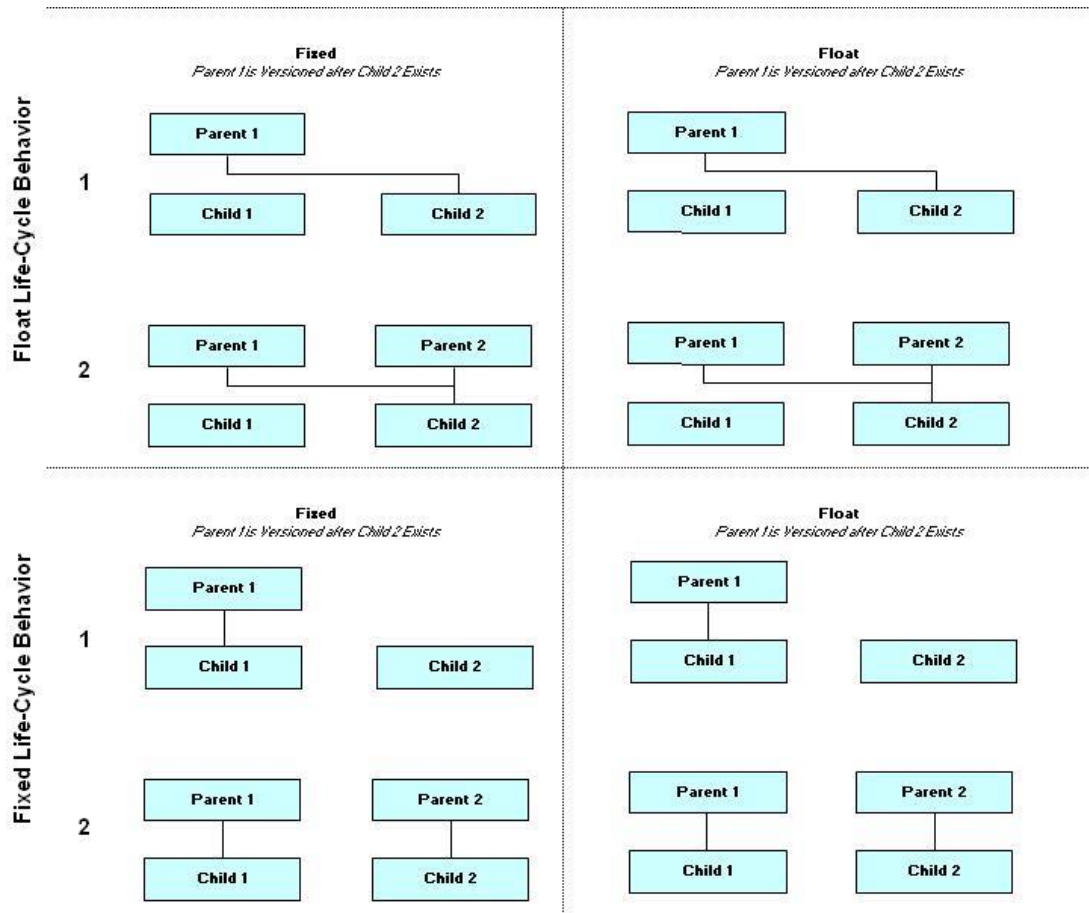
So far, we have been looking at all these cases with the assumption that the Parent or the source item is versioned first. Now, we need to examine all the same cases, but with the Child or the Related item versioned first.

4.2 Behavior When Child (the Related Item) Is Versioned First

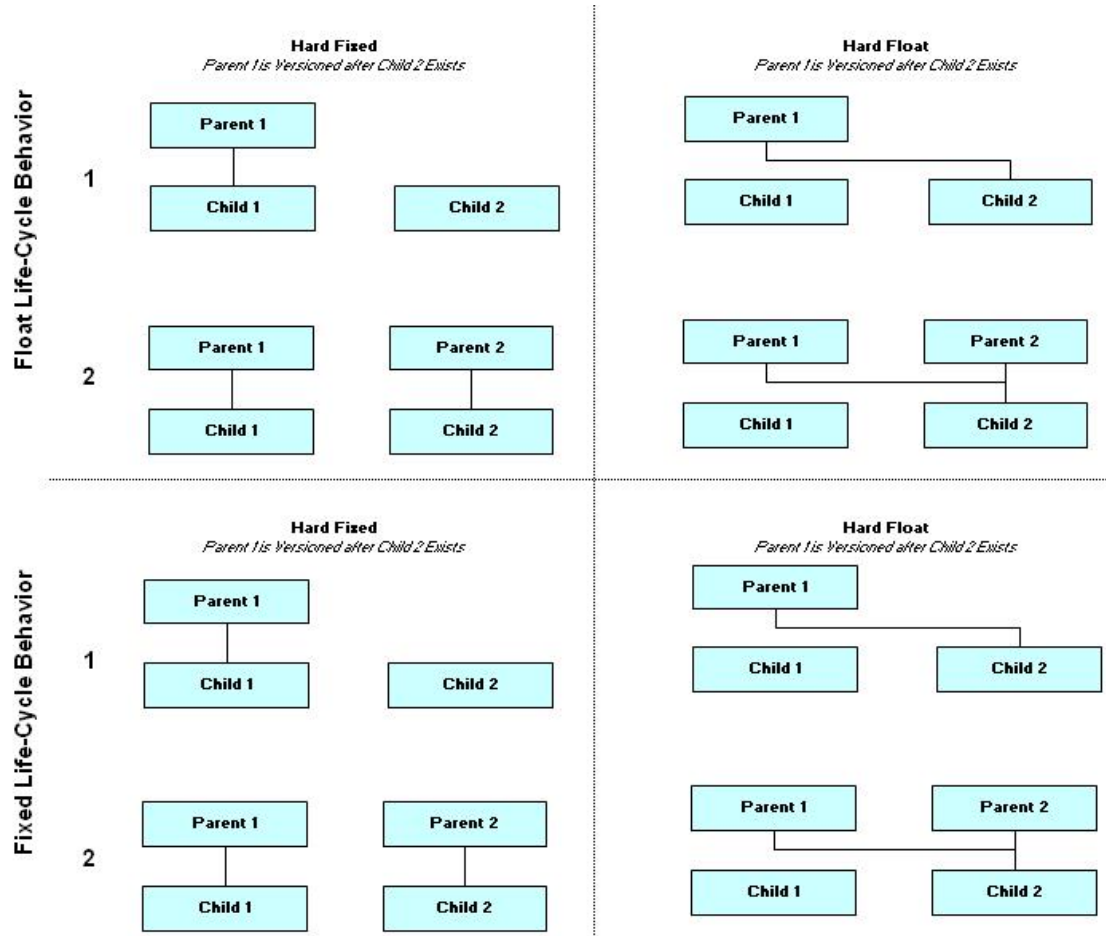
As previously, let's consider first what happens when there is no life cycle state behavior set. Notice that even for Fixed behavior, Parent 2 is associated with Child2. This is true because of the following rule: When a source item is versioned, it is automatically associated with the latest available version of all its children. In this case, since Child 2 was previously created, instead of pointing to Child1, Parent 2 points to Child2.



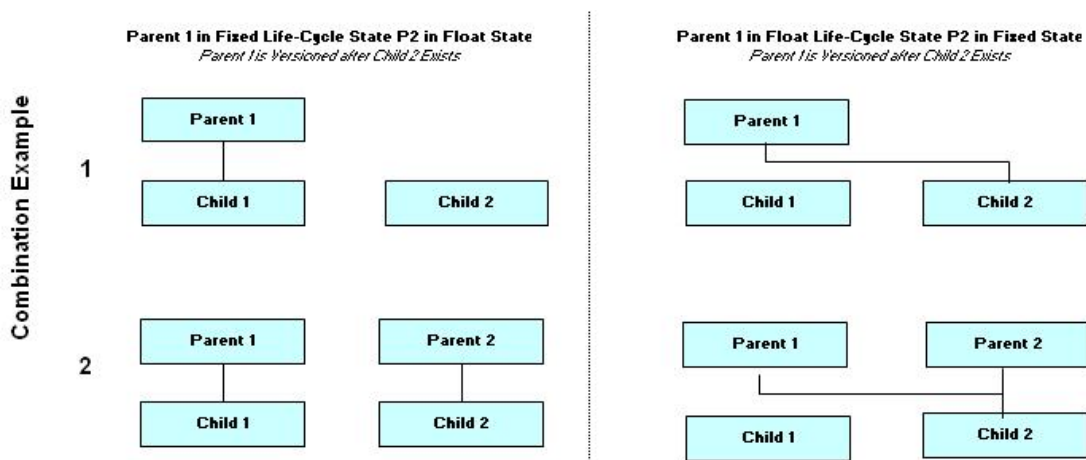
The Life Cycle State behavior overrides the RelationshipType behavior. In the diagrams below, Fixed relationship acts like Float when in Float life cycle state, and the Float relationship acts as Fixed in the Fixed life cycle state.



When the RelationshipType Behavior is set to Hard Fixed or Hard Float, it overrides the Life Cycle State behavior. In the diagrams, the Hard Fixed behavior acts as Fixed, regardless of the life cycle state setting, and the Hard Float acts as Float regardless of the life cycle state setting.



If the RelationshipType Behavior is not set to Hard, it can be overwritten by the Life Cycle State Behavior. The figure below shows what happens when one generation of the parent is in one state, with one type of behavior, and the next state is in another life cycle state, with a different behavior.



Remember that if the Life Cycle state behavior to Hard Fixed or Hard Float, it will override all subsequent state behavior through the end of the cycle. This is particularly useful if a life cycle map with states that follow Released, such as Obsolete, or Superseded. Most likely once an item reaches a Released state users would want to Hard Fix its behavior, so that its configuration remains the same, even if the item is promoted to Obsolete or Superseded.